Acta Technica *62* (2017), No. 5B, 1025–1032 © 2017 Institute of Thermomechanics CAS, v.v.i.

Bi-level genetic algorithm based on affinity propagation clustering for large scale traveling salesman problem

Yanyan Tan[1], Guoxiao Yun[1], Yanna Wu[1], Wei Zheng[1]

Abstract. As a novel clustering method, affinity propagation (AP) is quite different from $k-$means clustering, which needn't to artificially set the number of clusters in advance. In order to solve the large scale Traveling Salesman Problems (TSPs) efficiently, this paper presents a bi-level genetic algorithm based on affinity propagation clustering, abbreviated as APGA and compares it with classical genetic algorithm (CGA) and other clustering based algorithms. With affinity propagation clustering, APGA divides a large scale TSP into some subproblems, each sub-problem corresponds to a cluster. In the lower level, genetic algorithm is adopted to find the shortest hamiltonian cycle for each cluster. All those clusters can be handled in parallel. Then, all those clusters need to be linked into one whole tour by an effective connection method. Therefore, in the high level, a modified genetic algorithm is designed for integral optimization with the objective of shortening the whole traveling tour. At last, three sets of experiments on benchmark instances are carried out to verify the performance of the proposed APGA. Experimental results demonstrate its effective and efficient performance.

Key words. traveling salesman problem, large scale, affinity propagation clustering, genetic algorithm, bi-level.

1. Introduction

Traveling Salesman Problem (TSP) is a well-known combinatorial optimization problem, which belongs to the class of NP-hard optimization problems [1, 2]. This problem has attracted much attention of researchers and remains an active research [3-7], due to a large number of real-world problems that can be modeled by TSP.

TSP can be presented as follows: there are Ncities and distance matrix$D =$

[1] Acknowledgment - This paper is supported by the National Natural Science Foundation of China (Grant: 61401260, 61373081, 61402268, 61572298), and the Natural Science Foundation of Shandong China (Grant: BS2014DX006, BS2015DX016, ZR2014FM012).

[2] Workshop 1 - School of Information Science and Engineering, Shandong Normal University, Jinan, 250358, China

$(d_{ij})_{N\times N}$ which gives distances from one city to another city. A route can be seen as a cyclic permutation of cities from 1 to$N$. If $\pi(i)$is defined as the city visited in step$i$, $i=1,\ldots,N$, the cost of a route is presented as follows [1]:

$$\text{minimize } f_\pi = \sum_{i=1}^{N-1} d_{\pi(i)\pi(i+1)} + d_{\pi(N)\pi(1)} \tag{1}$$

The objective of the TSP is to get the tour with the minimum cost. Company with the exponential increase of the search space, large scale TSP is hardly solvable through finding the exact solution directly. Genetic Algorithm (GA) is a novel algorithm based on natural election for complicated problems with huge search, such as TSP. GA is effective for solving TSPs, nevertheless, with the number increase of the traveling cities, classical genetic algorithm performs poor effect. In order to solve the large scale TSPs efficiently, some researchers considered two-level genetic algorithm (TLGA) [8] or clustering based evolutionary algorithms and have obtained some promising results [2, 4, 9], in which genetic algorithm based on $k-$means clustering (KGA) is a well representative [4]. KGA performs effective for large scale TSPs, However, $k-$means clustering method is quite vulnerable to select exemplars initially, so it always need to optimize many times with different initialization and make the effort to obtain a preferable solution. Besides, the number of clusters needs to be set before running the algorithm. Therefore, it only performs well under some conditions that the amount of clusters is given and the situations are good that at least one random initial solution approach the good solution.

Affinity propagation clustering [10] is a novel clustering method. It can identify high-quality cluster centers by passing messages between data points, and needn't the number of clusters to be determined or estimated before running the algorithm. Up to now, AP has been used to handle a variety of clustering problems, such as identifying multispectral images, solving large immobile location–allocation problem and clustering in vehicular ad hoc networks. In order to high efficiently solve the large scale TSPs, this paper presents an improved bi-level genetic algorithm based on affinity propagation clustering (APGA), and compares it with other algorithms for verifying its performance. Experimental results on three sets of TSPs show its effective and efficient performance.

The rest content of this paper is organized as follows: Section 2 describes the affinity propagation clustering. Section 3 presents the proposed bi-level genetic algorithm based on affinity propagation clustering (APGA). Then in Section 4, experimental work and comparing results are provided. Finally, Section 5 concludes this paper.

2. Affinity propagation clustering

Affinity propagation (AP) clustering was first proposed by Frey and Dueck in Science, 2007. The main idea of AP clustering is to find the optimal representative point, called 'exemplar'. It is a fast clustering technology especially in the case of large number of clusters, and has some advantages, such as speed, general applicability and good performance. Different from $k-$means clustering, the AP method does not need specifying the initial cluster centers in advance. On the contrary, it regards all data points as potential cluster center, called 'exemplar', and iteratively exchanges messages between data points until a good set of exemplars and clusters emerges, therefore avoiding the arbitrary selection of the initial cluster centers.

Let$S\left(X_i, X_k\right)$ be the similarity between pointsX_i andX_k, i.e., the suitability of point $X_i$to serve as the exemplar for data pointX_k (shorten as $i$and$k$). In conventional AP, a common choice for similarity is the negative Euclidean distance

$$s\left(i,k\right) = -\left\|i-k\right\|^2 . \tag{2}$$

$s\left(i,k\right)$represents the similarity of points $i$and$k$for short. AP can use other more general notions of similarity, and the similarities may be positive or negative.

The preference of point$k$, called$S\left(k,k\right)$, is the apriori suitability of point $k$to serve as an exemplar. Preferences can be set to a global (shared) value or customized for particular data points. High values of the preferences will cause AP to find many exemplars (clusters), while low values will lead to a small number of exemplars (clusters). A good initial choice for the preference is the minimum similarity or the median similarity.

The AP algorithm is mainly described as follows [10], [15].

Step 1: Initialization

$$\begin{aligned} r\left(i,k\right) &= s\left(i,k\right) - \max_{k'\neq k}\left\{s\left(i,k^{'}\right)\right\} \\ a\left(i,k\right) &=0 \end{aligned} \tag{3}$$

Step2: Responsibility updates

$$\begin{aligned} r\left(i,k\right) &=s\left(i,k\right) - \max_{k'\neq k}\left\{a\left(i,k^{'}\right) + s\left(i,k^{'}\right)\right\} \\ r\left(k,k\right) &=s\left(k,k\right) - \max_{k'\neq k}\left\{a\left(k,k^{'}\right) + s\left(k,k^{'}\right)\right\} \end{aligned} \tag{4}$$

Step3: Availability updates

$$a\left(i,k\right) = \left\{ \begin{array}{l} \min\left\{0, r\left(k,k\right) + \sum_{i'\notin\{i,k\}} \max\left(0, r\left(i^{'},k\right)\right)\right\}, i\neq k \\ \sum_{i'\neq k} \max\left(0, r\left(i^{'},k\right)\right), i=k \end{array} \right. \tag{5}$$

Step 4: Making assignments

$$c_i^* \leftarrow \arg\max_{1\leq k\leq n} r\left(i,k\right) + a\left(i,k\right). \tag{6}$$

In the process, two kinds of messages are exchanged among data points, and each takes into account a different kind of competition. The "responsibility"$r(i,k)$, sent from data point i to candidate exemplar point k, indicates how well suited point i would be as a member of the candidate exemplar point k. The "availability"$a(i,k)$, sent from candidate exemplar point k to potential cluster members point i, indicates how the cluster $k$would represent point i. Responsibilities and availabilities are initialized as Equ.(3), and in the whole process, they follow the updating rules (4)–

(5). Messages are updated on the basis of simple equations searching for minima of an appropriately chosen energy function.

At any time during the clustering, the magnitude of each message reflects the current affinity of a data point to choose another data point as its exemplar. For point i, if point $k(k \neq i)$ maximizes$r(i,k)+a(i,k)$, then k would be considered as the exemplar of i, whereas $k = i$means that point i itself is an exemplar.

The message-passing procedure may be terminated after a fixed number of iterations, after changes in the messages are less than a threshold, or after the local decisions stay constant for some iterations.

3. Proposed bi-level genetic algorithm based on affinity propagation clustering

In order to high efficiently solve the large scale TSPs, this paper presents an improved bi-level genetic algorithm based on affinity propagation clustering (APGA). APGA uses affinity propagation clustering method to divide a large scale TSP into some sub-problems, each sub-problem corresponds to a cluster. In the lower level, classical genetic algorithm with ordinal encoding is applied to find the shortest Hamiltonian cycle for each cluster. All these clusters can be solved in parallel. Then, an effective link strategy is adopted to combine all those clusters into one with the object of optimizing the whole traveling circle. Different combing sequences among clusters will result in different traveling circles, searching for the shortest is our purpose. Therefore, in the higher level, an another modified genetic algorithm is designed for integral optimization with the objective of optimizing the whole traveling circle.

The integral procedure of the proposed APGA:

```
Input a TSP;
Step1: AP is adopted to cluster the TSP into$k$sub-problems;
Step2: For each sub-problem $i = 1$ to $k$, do:Repeat
Select parents for next generation;
Perform the crossover operator;
Perform the mutation operator;
Until stop criteria are met;
Output Hamiltonian cycle for sub-problem $i$; Endfor
Step3: Seek for the best combing sequence $S$ with GA;
Step4: Combine all those Hamiltonian cycles into one tour following the optimal sequence$S$;
Output the shortest whole traveling tour.
```

4. Experimental work

To investigate the performance of the proposed APGA, three sets of experiments on benchmark test instances [16] are carried out.

In the first set of experiments, three instances are used on affinity propagation algorithm for verifying its capability of clustering. Figure 1 illustrates the clustering results obtained by AP. It can be seen from the figure that, with the number increase of the travelled cities, the number of clusters divided by AP becomes large gradually, and the number of cities included in each cluster keeps not very large, which guarantees that those sub-problems can be easily solved by genetic algorithm.

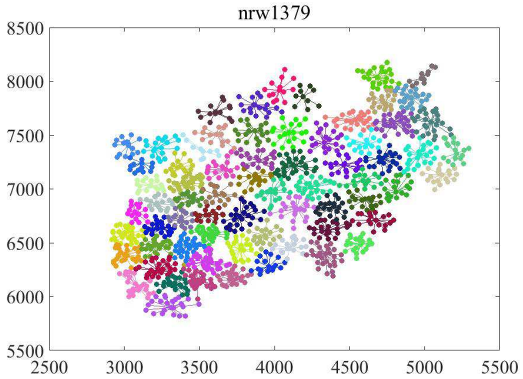

Fig. 1. AP clustering results of nrw1379, u2319 and pcb3038

Table 1. Comparisons of CGA and APGA

Problems	Optimum	CGA			APGA		
		mean	min	std	mean	min	std
pcb442	50778	$1.21*10^5$	$1.15*10^5$	279.3	$6.52*10^4$	$6.27*10^4$	257.8
att532	27686	$2.79*10^5$	$2.70*10^5$	263.2	$1.31*10^5$	$1.23*10^5$	156.2
d657	48912	$1.43*10^5$	$1.21*10^5$	43.5	$7.43*10^4$	$7.01*10^4$	47.9
rat783	8806	$3.82*10^4$	$3.48*10^4$	44.6	$1.81*10^4$	$1.46*10^4$	39.6
rl1889	316536	$4.02*10^6$	$3.82*10^6$	297.4	$1.76*10^6$	$1.53*10^6$	1807.3
fl3795	28772	$1.61*10^5$	$1.52*10^5$	412.7	$1.03*10^5$	$9.87*10^4$	1066.5

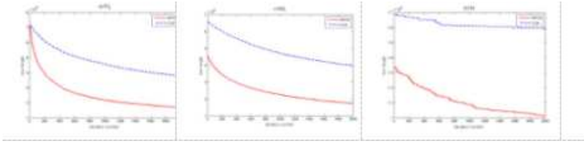

Fig. 2. Convergence curves of APGA and CGA on rat783, rl1889, fl3795

Table 2. Comparisons of KGA and APGA

Problems	Optimum	KGA			APGA		
		mean	min	std	mean	min	std
pcb442	50778	62129.1	61461.5	279.3	61735.6	61242.2	247.1
rat575	6773	7647.09	7820.56	263.2	7631.32	7621.47	150.8
d657	48912	56785.3	56738.2	43.5	56801.5	56784.1	27.9
rat783	8806	9882.9	9822.0	44.6	9747.2	9632.8	35.3
u2152	64253	75689.3	75184.4	297.4	75571.8	73825.2	180.3
u2319	234256	243336.3	242605.5	412.7	245704.4	249980.1	326.5
pcb3038	137694	159777.9	159182.2	906.7	155086.5	150264.2	512.4
rl5915	565530	786908.3	780619.9	7590.9	656647.9	619020.9	5895.5

In the second set of experiments, six instances are carried on APGA for testing its performance. The results obtained by APGA are compared with those obtained by classical genetic algorithm (CGA). Population size and evolution generations are set to 100 and 20 respectively. Table 1 presents the experimental results for each benchmark problem and statistics for 20 independent runs. Figure 2 shows the evolution of the tour length with the number of generations on three large scale test problems. It can be seen from Table1 and Figure 2 that, APGA performs much better than CGA on all these TSPs, which demonstrates that clustering-based algorithm is efficient.

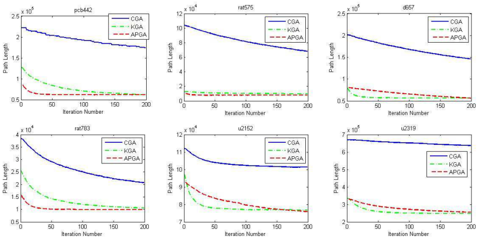

Fig. 3. Convergence curves of APGA and CGA on the second three test instances

In the third set of experiments, eight traveling salesman problems are tested on CGA, APGA and bi-level genetic algorithm based on k-means clustering (KGA) [16] for demonstrating the effectiveness of affinity propagation clustering. Population size and evolution generations are set to 200 and 200 respectively. All these three algorithms stop running till the stopping conditions are satisfied, i.e. the generations reach the max bound 200.

Table 2 presents the experimental results for each benchmark problem and statistics for 20 independent runs. As listed in Table 2, except for d657 and u2319, the mean and min values obtained by APGA are respectively smaller than those obtained by KGA for the other six test problems, which indicates that APGA performs better than KGA on almost all these TSPs. Meanwhile, the standard deviation (std) statistics obtained by APGA are also smaller than those got by KGA, which demonstrates that affinity propagation clustering performs robust than $k-$means clustering. Generally speaking, APGA performs a bit better than KGA in optimizing the TSPs. Meanwhile, $k-$means clustering is quite sensitive to the initial values of centers, and the number of clusters needs to be set in advance. Then affinity propagation clustering is preferable in bi-level genetic algorithm based on clustering for solving traveling salesman problems.

Figure 3 shows the evolution of the tour length with the number of generations. Comparing to CGA, KGA and APGA show obvious advantages over the ordinary algorithm. Meanwhile, APGA shows superior to KGA on pcb442, rat575, rat783. On u2152, APGA and KGA perform equally. Both APGA and KGA can obtain superior initial solutions, which owes to clustering methods. These results indicate that the algorithms based on clustering converge much faster than CGA. From Table 2 and Figure 3 we can conclude that KGA and APGA are more efficient and effective than CGA, they perform well on getting more reasonable tours under limited time for large TSPs. On the other hand, affinity propagation clustering is preferable in clustering based bi-level genetic algorithms for solving TSPs.

5. Conclusions

In order to high efficiently solve the large scale TSPs, this paper presents an improved bi-level genetic algorithm based on affinity propagation clustering (APGA), and compares it with classical genetic algorithm (CGA) and KGA. With affinity propagation clustering, APGA divides a large scale TSP into some sub-problems, each sub-problem corresponds to a cluster. In the lower level, genetic algorithm is adopted to find the shortest hamiltonian cycle for each cluster. All those clusters can be handled in parallel. Then, all those clusters need to be linked into one whole tour by an effective connection method. Therefore, in the high level, a modified genetic algorithm is designed for integral optimization with the objective of shortening the whole traveling tour. At last, three sets of experiments on benchmark instances are carried out to verify the performance of the proposed APGA. Experimental results demonstrate its effective and efficient performance. Comparing results with other algorithms demonstrate that APGA is competitive and preferable.

References

[1] G. Gutin, A. Punnen: *The Traveling Salesman Problem and its Variations.* Kluwer Academic Publishers (2002).

[2] L. Tan, Y. Y. Tan, G. X. Yun, Y. N. Wu: *Genetic algorithms based on clustering for traveling salesman problems.* In: Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (2016), 103–108.

[3] C. C. Lo, C. C. Hsu: *Annealing framework with learning memory.* IEEE Transactions on Systems *28* (1998), No. 5, 648–661.

[4] Y. Tan, L. Tan, G. Yun: *Bilevel Genetic Algorithm with Clustering for Large Scale Traveling Salesman Problems.* In: International Conference on Information System and Artificial Intelligence (2016).

[5] H. K. Tsai, J. M. Yang, Y. F. Tsai: *Heterogeneous election genetic algorithms for traveling salesman problems.* Engineering Optimization *35* (2003), No. 3, 297–311.

[6] M. Dorigo, L. M. Gambardella: *Ant colony system: a cooperative learning approach to the traveling salesman problem.* IEEE Transactions on Evolutionary Computation *1* (1997) 53–66.

[7] J. L. Liu: *Rank-based ant colony optimization applied to dynamic traveling salesman problems.* Engineering Optimization *37* (2005), No. 8, 831–847.

[8] C. Ding, Y. Cheng, M. He: *Two-Level Genetic Algorithm for Clustered Traveling Salesman Problem with Application in Large-Scale TSPs.* Tsinghua Science and Technology *12* (2007), No. 4, 459–465.

[9] A. F. EI-Samak, W. Ashour: *Optimization of traveling salesman problem using affinity propagation clustering and genetic algorithm.* Journal of Artificial Intelligence and Soft Computing Research *5* (2015), No. 4, 239–245.

[10] B. J. Frey, D. Dueck: *Clustering by passing messages between data points.* Science *315*, (2007), No. 972, 972–976.

Received November 16, 2017